REMARKS

Claim rejections 35 USC § 102

Rejection of claims 1-17 as being anticipated by Flockhart et al.

Applicant has carefully considered Examiner's comments as set out in section 5 of the Office Action. In view of Examiner's suggestion with regard to the clarity of the independent claims, Applicant has replaced the term '*reference*' with the term '*pointer*' throughout the claims.

Claims 1 – 17 now require that each software object representing a contact comprises a pointer to at least one other software object.

Applicant submits that this clarifying amendment substantiates the arguments presented in the previous response of 10 October 2008.

In particular, the tech-support skill-value queue of Flockhart comprises a plurality of data entries, each data entry corresponding to a call and having an associated priority. The system of Flockhart is arranged to traverse from the head entry down through the list of data entries in the queue to identify the first occurrences of each priority level.

In contrast with the software objects of the present invention, these data entries simply represent information associated with the calls, i.e. an identification of the call and an associated priority level, (col. 5, lines 3 to 5). They do not comprise any information for directing the flow of the process. On the contrary, the system of Flockhart is simply designed to parse the queue in a predefined order, i.e. in the order of their arrival in the system (col. 4, line 64 –65, and col.5 43 to 46).

The "software object" as claimed in claims 1 to 17, is a more sophisticated entity arranged to hold links or pointers to other software objects.

This unique feature of the present invention enables each software object associated with a contact to maintain a link to at least a second software object associated with a second contact immediately ahead and/or behind the contact in the queue. In this way, the requirement for a traditional queue of the type disclosed in Flockhart in which the contact manager is required to monitor the list of objects is dispensed with, as each software object "knows" its place in the queue.

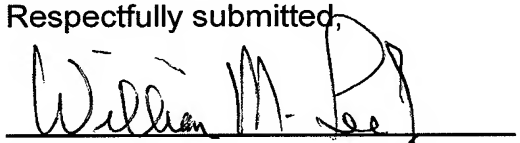
The present invention provides an aggregate collection of software objects having queuing information built into them by means of the pointers held therein. Thus, the invention allows one to inherently provide for a prioritised queue simply by maintaining a collection of software objects each of which points to another of the software objects. In other words, the collection of software objects in the present invention effectively is the queue.

Such a collection is stateless, unlike Flockhart's queue and is therefore more robust and failure resistant.

In view of the amendments and arguments made herein, Applicants respectfully request that the examiner withdraw the rejections, and allow the application.

April 2, 2009

Respectfully submitted,



William M. Lee, Jr.
Registration No. 26,935
Barnes & Thornburg LLP
P.O. Box 2786
Chicago, Illinois 60690-2786
(312) 214-4800
(312) 759-5646 (fax)